

My Second Bike: A TV-enabled social and interactive riding experience

Jaewoo Chung*, Kuang Xu*[†], Andrea Colaco*, Chris Schmandt*, Victor O.K. Li[†]

*Media Laboratory, Massachusetts Institute of Technology

[†]Department of Electrical & Electronic Engg., The University of Hong Kong

Abstract—In this paper, we propose a novel concept for a social TV application targeting the demographic of viewers enjoying live sports events, such as road bicycle racing. We intend to enhance the viewing experiences of spectators with sensor-fitted bikes tied to an interactive biking environment on television. The system enables a new form of personalized, physical, and virtual-reality interaction between viewers and a TV program, as well as interactions within or between communities of friends. We also describe a prototype we have implemented to demonstrate the feasibility of our idea. The prototype, My Second Bike, uses a 3D mirrored world environment (Google Earth) to visually represent participating spectators, competing athletes and outdoor bikers. We contend that the system has the potential to attract and support a large user base on account of its scalability, ease of deployment and ability to promote audience participation in live sports events on TV.

I. INTRODUCTION

Shared experiences around a television have been a driving force for co-viewing content. Sport-related events, moreover, tend to create a huge impact, bringing together large audiences to participate both physically and socially. People engage in sports for recreation, exercise, and also as spectators. Very often spectators vicariously participate in the experiences of the athlete by following the event closely - either live or on screen. It is not uncommon to find oneself imitating the actions of an athlete on screen or even repeating the lines of a favorite sitcom character. These aspects of television-related content have potential to be channeled into areas that can benefit from sustained engagement and interactivity. My Second Bike is a platform to create a social riding experience by providing a meeting space for bikers physically riding for recreation, bikers competing in sporting events and bikers exercising at home. The platform provides each rider a representation on screen and visual cues of nearby riders. By creating a sense of participation via direct visual/audio feedback and co-presence, My Second Bike aims to have a reinforcing effect in achieving sustained participation. Drawing from the positive effects that competitive gaming and sports events have, My Second Bike supports riders challenging each other to races and competitions. The framework allows biking related and presence information of real-world riders to be fed into the screen environment. This allows home-riders to challenge their real-world counterparts to biking races or even an exploratory ride. To capture the strong element of enjoying a competitive sporting event on television and creating a feel of participating in it, My Second Bike has a provision for riders to navigate

the terrain a bike-racing team may currently be climbing. All these aspects of the system can potentially create a community of bikers each with their own preference for riding all linked by a common platform.

Further, the system is developed using easily accessible 3D environments such as Google Earth, rendering it scalable and deployable to large audiences. This could potentially create similar effects as gaming consoles and merge into one application the key sociability and collaborative aspects of gaming with television screens - a sociable riding experience. One can imagine a typical user scenario using this system as elaborated below:

Matt is a high school student who commutes to school by bike everyday and likes hiking during the weekends with his friends. Today, he is watching TV featuring the Tour de France. He opens a gadget in the TV, which lists a menu to choose which riding group he would like to join. Matt selects his school's community group that his friends have just joined. On the left of the TV screen, he sees video streaming from France and on the right half of the screen, he sees a mirrored world of 3D model with bike avatars that represent his bike, athletes' bikes and others who are currently part of the group. He also recognizes other bikes that represent a nearby rival school's biking community. Not only competing with the athletes and participants of his group, but Matt is also competing as a team with other groups. Matt turns on the device attached on his bike (mounted on a bicycle trainer), which detects rotation of his back wheel and the angle of the handle. The device is connected to his TV via Bluetooth automatically. The viewers are not simply enjoying the Tour de France, but also participating in the very activity they are watching on television. They're also able to connect and share this experience and activity with members of their social networks and online communities.

In sections to follow, we describe background work in this area (Section II) followed by system overview (Section III) and implementation details (Section IV); we also discuss key features different from existing interactive applications and limitations of our system (Section V). We conclude with possible additional useful features to the current system (Section VI).

II. BACKGROUND

There has been an impressive surge in social and interactive TV applications. The television experience has shifted from

being a passive medium for distribution of professionally authored narrative-rich content to a combination of narrativity and interactivity as well as the ability to manipulate content. Further, opportunities for interacting with screen content and people co-viewing the same content are increasingly being provided. In many such applications common forms of engaging viewers include voting on a topic, conversations via instant messaging on issues emerging from on-screen content, awareness of remote viewers and discussions in online communities. With these trends in place viewers are increasingly interested in interacting with their screen content and as a result a number of new social television applications have been seen in the market in the past few years.

2BeOn [6] is an interpersonal communication service for TV users to communicate with their peers watching the same TV show as themselves. It was an attempt to reinforce socialization between viewers and took detailed account of the limitations of TV as an output terminal as well as the role of content in promoting conversations. Along similar lines, Reality Instant Messaging [9] provides services for conversations that involve groups of users added to a bot interface rather than focusing on desktop-like single user applications. Additionally, it also ties this social network service with real-time sports events (in their example, golf). The system sends out synced information about events in the game and participants can discuss these via the bot interface. Cabo Interactiva [11] in Portugal offers a number of interactive TV applications including shopping, banking, communicating and sport-related applications. Most popular among these was their Sports TV channel's interactive football that offered video stream selection, participating in debates and polls, instant shopping for football products, etc. It turned out that the forums witnessed a lot of activity, with active comments, cheering and competition among supporters of opposing teams. Similarly, many other solutions have been proposed and implemented to capture and represent a person's social network while watching television, sharing multimedia messages, finding friends and their favorite programs and interacting while viewing content.

Several TV applications specifically based on live sports events have been proposed. [14] aims at enabling real-time interaction between audiences and the video director, so that the video director can schedule future broadcasts based on viewers' feedback about which parallel sports event they enjoy more. [10] endeavors to facilitate personalized video delivery by enabling a viewer to customize the live sports video according to his own interest. Based on multi-camera angle selection, athletes/objects matching the personal preferences can be automatically tracked. To enhance the collective viewing experience, [12] further proposes a model that learns and develops rules for group preferences by clustering, improving social communication and sense of belonging to a group, among spectators. However, our work is different in that, we intend to engage viewers to physically interact with and participate in a TV program, thereby increasing the sense of recreational and social connectedness among them.

On the other hand, the concept of using a real bike as

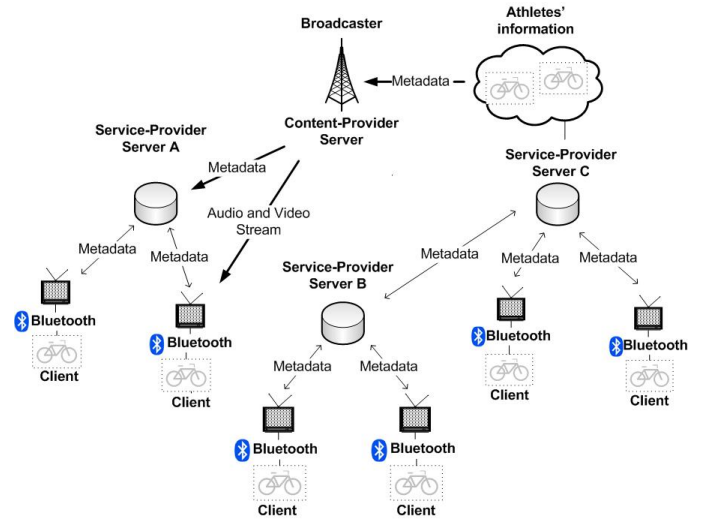


Figure 1. Topology of data communication architecture between content-provider, service provider and clients of TV viewer.

controller of a virtual bike in Virtual Reality environments has been explored many times during the last decade [7] [8] [13]. In recent years, many console based arcades and on-line games already have made sophisticated multi-user based interactive racing games, but, to date, we have not encountered a project that utilizes the two parts together – interactive TV and virtual reality games – to allow viewers to participate in the same activity as the live event being broadcast through TV channels e.g, live sports games.

III. SYSTEM OVERVIEW

Implementing the system described in the scenario (in Section I) requires the following three main components: content-providers (or broadcasters), service-providers, and clients (or content consumers). The main task for the broadcaster is to send two types of information, video and audio (TV content), and metadata that is associated with the content. In our case, we simulated the data portion that is streaming the meta information to viewers via IP. A Service provider server serves as an aggregator of metadata from an event broadcaster and registered users. The gathered data will be re-distributed to the registered users so that the client side at each user can visualize all the registered users' and athletes' bikes. The main functions for the clients are to exchange metadata with the server and visually represent this metadata. In detail, the tasks are: *i*) collecting control data from the bike via a device that detects rotation of the back wheel and heading angle from the handle, *ii*) generating and sending out metadata associated with a given user's bike, and *iii*) based on the received streams of metadata from the Service-provider Server, simulating and visually representing biking participants on TV. This visualization serves as a direct feedback of the user's performance compared with other avatars' relative positions and performance on the TV screen.

As Fig. 1 depicts, the network connections between server-server and server-clients are essentially HTTP requests over the Internet. Between the content providers and server-client



Figure 2. An evaluation scenario of the system prototype.

sides, the data network can be one of many options viz. satellites, TV cable lines, Internet infrastructure, etc. that support transmitting metadata from broadcasters. Multiple servers can host clients, and each server can connect to any numbers of peer servers and exchange metadata over the Internet by opening a socket connection - an example connection is shown between Server B and Server C in Fig. 1. The athletes' data can be collected via short distance radio transmitted from a GPS equipped device to nearby outdoor broadcasting van that is following the athletes. Alternatively, a GPS equipped phone can transmit GPS coordinates, Athletes' IDs, and heading information to content providers over GPRS. In our case, we used the latter method.

IV. IMPLEMENTATION

We have implemented a working prototype of our proposed system. The prototype demonstrates the core functionality of the system for multiple users. It also highlights the feasibility of deploying and scaling this system. Fig. 2 shows an evaluation scenario of the system prototype, in which two indoor viewers ride their own bikes enjoying the same TV program of road bicycle racing while physically based in different places. On the TV screen, the left part displays the TV program and the right part shows the presence of viewers and athletes as avatars in a 3D mirrored world utilizing Google Earth's API.

The main components that make up the system include the Content-provider Server (Broadcaster), Service-provider Server (Application server), Clients (content consumers) that consist of a desktop computer connected to an HDTV, and sensor-fitted Bikes. In the following we describe these parts in detail.

A. Content-provider Server (Broadcaster)

The Broadcaster in My Second Bike is the television content provider - any digital TV broadcaster could potentially function as a content-provider server. In the prototype, we equip the bike of an outdoor biker with a cell phone (Motorola i870) with Assisted Global Positioning System (A-GPS), which provides not only the GPS coordinates, but also the heading information. We have implemented a lightweight J2ME application that runs as a background process to

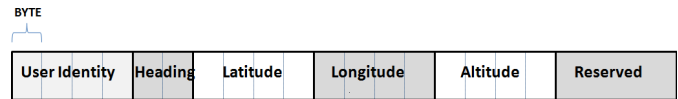


Figure 3. The metadata object.

retrieve GPS coordinates periodically to gather information for generating metadata, which contains essential information for our system to work. Fig. 3 depicts the content and format of the metadata. During our experiment, we observe the Altitude item in the metadata is inaccurate as provided by GPS; so we do not transmit this information through the network, and we calculate it at the client side.

The application sends the metadata to the Content-provider Server via UDP (User Datagram Protocol) over Integrated Digital Enhanced Network (iDEN) [2]. During the experiment, we observe the average delay for data transmission was around 220ms. The Content-provider Server listens to this incoming metadata and synchronously passes this information on to any Service-provider Server connected via TCP/IP over the Internet. In our implementation of the Content-provider server, we have omitted the transmission of video and audio.

B. Service-provider Server

The Service-provider Server hosts a database of real-time metadata, serving as an intermediary between different clients, and between clients and Content-provider Server. We utilize Java Servlets to realize the server-side function: a TCP daemon is set up to initiate connection to the Contentprovider Server and to listen for incoming metadata from the Content-provider. For communication with clients, we utilize the Direct Web Remoting (DWR) [1] toolkit which supports the reverse-AJAX feature. We use the Piggyback mode such that when a client browser requests to upload its status to the Service-provider Server, the server sends the most recently updated metadata of other clients along with the response that the client is expecting. This also reduces unnecessary network transmission cost and eases the server load. Metadata received by the server (from both content provider and clients) is stored in a database and communicated to intended clients.

To minimize network resource consumption, clients can filter metadata information of other participants by setting proximity boundaries - the server would then have to send metadata of clients within a boundary specified. Additionally, there is an audio component in this system design which sends out audio alerts informing clients about bikers in their vicinity, as well as updates about their performance relative to competing bikers.

Before exchanging metadata between the Service-provider Servers, each server computes metadata that holds an average of the locations of the connected clients. This allows each server to not only reduce the amount of metadata but also make a Service-provider Server represent a particular team of participants. Thus, participants can compete not only individually but also as a team member of a community, e.g. my high school team represented by a Service-provider Server or nearby rival school server. This promotes each server to serve

as a local/online community. In practice, with large-scale user participation, this community-based service can be hosted by any social networking website such as Facebook.

C. Client

The client consists of a desktop computer connected to an HDTV. It uses a 3D model rendered by Google Earth. The client requires a browser that is compatible with Google Earth API, supports JavaScript and OpenGL. The Google Earth model is linked to our Service-provider Server and presence of riders is updated and visually represented as avatars in the rendering. Using Google Earth has many advantages, important among them are:

- reasonably accurate 3D models - thereby offering a huge number of places for users to choose from.
- open source API and hence customizable and easily accessed by clients at no cost.
- dynamic and continuous refreshing of the 3D model, which can be exploited by other components of the system to update location information.
- large user base.
- ease of scalability - even a large number of clients concurrently accessing the system does not overload our Service-provider Server as they are handled directly by the Google Earth server.

Further, both Google Earth and our Service-provider Server use AJAX to update client information and synchronize information between clients. This is an added advantage to our system in that clients do not have to make a request to the server and wait for a response; rather, the server asynchronously sends client information whenever available and necessary. The metadata is communicated between client and server using JavaScript Object Notation (JSON) format. These JSON packets being object-based are easily re-usable and save sending multiple packets in response to multiple requests. It also aids in solving multi-client synchronization since the metadata of all the involved clients are packed in one JSONArray for transmission. In the prototype, each client uploads its metadata periodically (every half second).

D. Sensor-fitted Bike

The Smart Bike in our system is an instrumented bike retrofitted with sensors and encoders, capable of communicating information to our Service-provider Server. The connection between the Client (simulated in desktop PC) and sensor-fitted Bike is through virtual COM-port which is created from Bluetooth-pairing between bike and TV client in Microsoft Windows OS. However, communication between COM port and web browser using JavaScript is not usually supported. Our solution to this is to utilize ActiveX to tunnel a channel that allows JavaScript to communicate to COM port [4]. This enables receiving sensor data by web browser with JavaScript that allows visualization of Google Earth 3D model and generates metadata for sending it to the Service-provider every second.

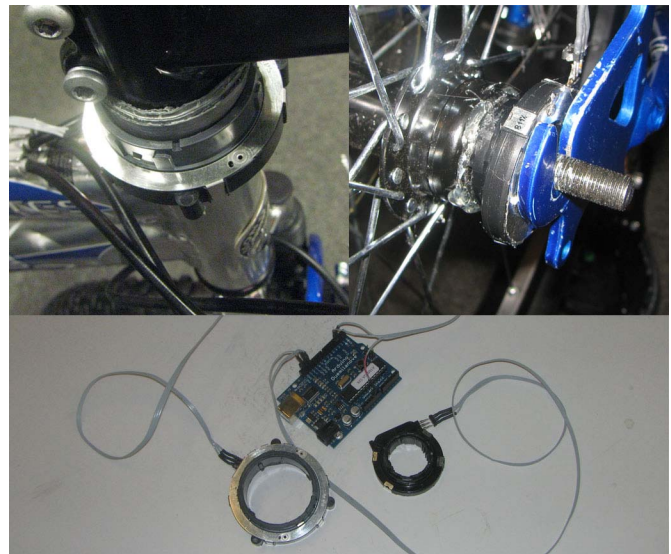


Figure 4. The installation of powerless rotary encoders on a bike.

In the prototype, the bike is mounted onto a bicycle trainer so that a user can ride it in front of the TV screen. The requirement is that the bike should be able to continuously monitor its motion status, such as heading direction and distance traveled, as if the user is riding in the physical outdoor scenario, and upload them to the Service-provider server in real time. We utilize two powerless 60-step rotary encoders as sensors that are wired to a button-battery powered micro Bluetooth controller on the bike. One sensor is mounted on the shaft of the bike knob, to detect the angle of the knob when a user makes a turn. Another one is installed on the hub of the bike's back-wheel, which is responsible for reading the rotation status of the wheel. Our rotary encoders have 60 steps granularity that provides resolution of 6 degrees. The micro-controller board collects the sensor data every millisecond. It pre-processes this data to calculate the heading direction and speed of the bike, and transmits the result to the TV client through Bluetooth radio every 30 milliseconds, so that the rate of updating user status is higher than the frame refreshing rate of HDTV. We find that the 6 degree resolution works well for detecting both handle-turning and wheel-rotation. Fig. 4 shows the installation of sensors on a bike. In practice, if we customize the control-board and utilize standard UART port instead of Bluetooth for data communication between bike and TV client, the cost of manufacture is less than \$20.

V. DISCUSSION

My Second Bike has proposed and prototyped a new paradigm for social interaction with televisions. Watching sports-related television content is an inherently social activity that engages large audiences and encourages participation. Our implementation, as described, leverages these aspects of sports-spectating. Therefore, we anticipate the system to increase the sense of social connectedness among viewers, promote an immersive experience and better represent remote presence in television environments. This system differs from

current forms of viewing and has potential to support a large user base and sustain interactivity.

Additionally, using Google Earth as a 3D riding environment can also contribute positively to the user experience as it mirrors the real world in contrast to other virtual environments such as Second Life. The familiarity with a particular real world space may make the navigating experience easier and more engaging. Alternatively, an unknown area that maps to an interesting area in the real world could also engage users to undertake exploratory challenges while riding.

However, in implementing My Second Bike some technical challenges were encountered that could impact the overall user experience as well as scalability.

- *GPS accuracy.* Imprecise GPS coordinates can result in misplacement of avatars in the 3D model, which may place the avatar off the road. This problem can be overcome by snapping GPS coordinates to the race course which would place avatars correctly on the riding path.
- *AJAX same origin policy.* Due to security (data confidentiality and integrity) requirements imposed by modern Web-browsers, certain elements such as XMLHttpRequest object in AJAX suffer from the same origin policy which prevents the application from connecting to domains different from its domain of origination. However, there are several effective ways of coping with the same origin policy, such as, installing a proxy on the consumer Web server [5], utilizing dynamic *<script>* tags at the client side, configuring Apache's *mod_rewrite* property at the server side, or modifying the preference options of client browser (on the trade-off of data security).
- *Network cost.* The Service-provider Server broadcasts metadata of all the athletes to all the spectators, which would not scale well with a large number of participants. This can be countered by providing a dynamic profile-matching mechanism for the system so that participants are provided metadata only of athletes they choose to ride with.

VI. CONCLUSION

In this paper, we have developed a prototype, My Second Bike, to actively engage television spectators in on-screen sports events. My Second Bike promotes a shared riding experience with athletes and outdoor bikers by providing a shared riding space in 3D mirrored worlds such as Google Earth. By introducing an option for community riding and visually representing riders, this system can create a social experience while riding even in your living room. The current phase of implementation of My Second Bike offers a platform for development of new features that can improve user experience.

- A desirable feature could be superimposing users's bike avatars on live TV video streams using Augmented Reality (AR) technology. Having visual feedback by fitting one's avatar directly in the TV video stream may create a deeper sense of participation in the activity.
- This system can be extended to track and field athletics, swimming, car racing etc. For such activities, we can

envision a virtual "track/lane" for the participant by applying AR technology to draw this track and represent the avatar directly in the environment.

- Riding My Second Bike can be made a more real experience by transmitting sensory effects to participants, e.g, force feedback, variable resistance to represent uphill and downhill riding conditions, rough and bumpy terrain etc. MPEG RoSE [3] is a potential candidate that could fit in our system framework to use sensory information to actuate appropriate physical sensors to enhance user experience.
- Creating an audio and video channel between participants could be another interesting feature. Our current system can be used for team races such as relay-riding. This form of riding would require participants to communicate with each other as a rider takes over the previous virtual rider. Audio communication would prove very valuable in this case. It would also promote interactivity among participants. Further, it could give a better sense of distance between virtual participants by providing distance-based volume control.

ACKNOWLEDGMENT

We thank Dr. Ig-Jae Kim of MIT Media Laboratory for his comments throughout the paper. We are also grateful for the support received from Samsung Electronics Co., Ltd.

REFERENCES

- [1] Direct web remoting. <http://directwebremoting.org/dwr/index.html>.
- [2] <http://webaugur.com/matt/files/nextel/techover.pdf>.
- [3] <http://www-itec.uni-klu.ac.at/~timse/research/publications/rose.pdf>.
- [4] <http://www.activexperts.com/activcomport/howto/html/>.
- [5] Javascript: Use a web proxy for cross-domain xmlhttprequest calls. <http://developer.yahoo.com/javascript/howto-proxy.html>.
- [6] J. Abreu, P. Almeida, and V. Branco. 2beon: interactive television supporting interpersonal communication. In *Proc. of the Sixth Eurographics Workshop on Multimedia 2001*, pages 199–208, 2002.
- [7] D. C. Brogan, R. A. Metoyer, and J. K. Hodgins. Dynamically simulated characters in virtual environments. *IEEE Comput. Graph. Appl.*, 18(5):58–69, 1998.
- [8] G. U. Carraro et al. The peloton bicycling simulator. In *Proc. of the 3rd Symposium on Virtual Reality Modeling Language*, pages 63–70. ACM, 1998.
- [9] M. Chuah. Reality instant messaging: injecting a dose of reality into online chat. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems*, pages 926–927. ACM, 2003.
- [10] S. Poslad et al. Directing your own live and interactive sports channel. *International Workshop on Image Analysis for Multimedia Interactive Services*, 0:275–279, 2009.
- [11] C. Quico. Are communication services the killer applications for interactive tv? In *Proc. of European Conference on Internet Television: from Viewers to Actors*, 2003.
- [12] Z. Wang, S. Poslad, and A. Pearmain. A collective director for highly interactive viewing of live sports events. In *Adjunct Proc. of EuroITV 2009 Networked Television*, 2009.
- [13] R. C. Waters et al. Diamond park and spline: Social virtual reality with 3d animation, spoken interaction, and runtime extendability. *Presence*, 6(4):461–481, 1997.
- [14] C. M. Williams and R. Wages. Video conducting the olympic games 2008: the itv field trial of the eu-ist project live. In *Proc. of the 3rd International Conference on Digital Interactive Media in Entertainment and Arts*, pages 436–440. ACM, 2008.